Sentiment analysis for users' reviews

Zhe Hu Yu Yin Shaoyang Yan

0 Overview

Shopping on-line is becoming increasingly popular among the world now, and more and more people prefer to shop on-line. Under this situation, understanding customers' comments become key to analyze a user's likes and dislikes. One of the essential method is sentiment classification. In this project, we build several models to classify the users' reviews being positive or negative based on supervised and unsupervised learning approaches. Also, we design the different models for both English and Chinese data.

1 Dataset and Data Cleaning

Our project uses two datasets. One is English dataset from UCI web consisting 3000 comments among Amazon, Yelp and IMDb. The other dataset is in Chinese, which is the comments about a kind of water healer from Jing Dong on-line market. We use open source data on the Internet.

Firstly, delete the duplicated sentences in both positive comment set and negative comment set. This is because many people just copy other people's comments and such duplicated comments are meaningless and If there are too many such duplicated comments, this will mislead the classification. Then delete the duplicated part in each sentence due to the same reason. After the processing, we split the comments into words. For English data, there are about 5000 words that have shown up in training set. Besides, words such as "I, that, the" do not contain important significance to be used in sentiment analysis. Hence, we have to filter out the stop-word in each comment.

As for Chinese data, it's much more difficult to process due to the complexity of structure and grammar in Chinese. There are no blank between each word. Word segmentation tools to separate the sentences into single words. There are about 19000 words that have showed up in the training set and many occurs only few times. Using all words to classify sentence will cost lots of time. We choose those words whose frequency is above 50. And the number of words whose frequency beyond 50 is 1094.

Thus far, the words extracted from comments can be used for constructing a word vector. In the vector, 0 represents that the feature appears in the document, while 1 represents the feature not appearing in the document.

2 Lexicon-based methods

Lexicon-based methods for sentiment analysis is a simple and conditional way to extract sentiment from text, which is a simulation of how human beings remember and use their obtained information to classify a sentence from good to bad. Lexicon is built by assigning words like “love, enjoy, great…” as positive values, and words like “hate, dislike, bad…” as negative values. Except for polarity, the strength of each word in the lexicon is also given by considering the frequency and other features. [1] After having the lexicon, we preprocess the data and compare words in target sentences with in lexicon, then classify the target sentences by summing up the total scores of each sentence. If the whole customer review has an over zero score, we classify it as a positive sentiment, which means this customer likes the product. Otherwise, negative and this customer will probably not recommend this product to his or her friend.
The whole procedure can be denoted as Fig. 2.1. The input dataset is downloaded from the UCI web[2]. Data preprocessing includes

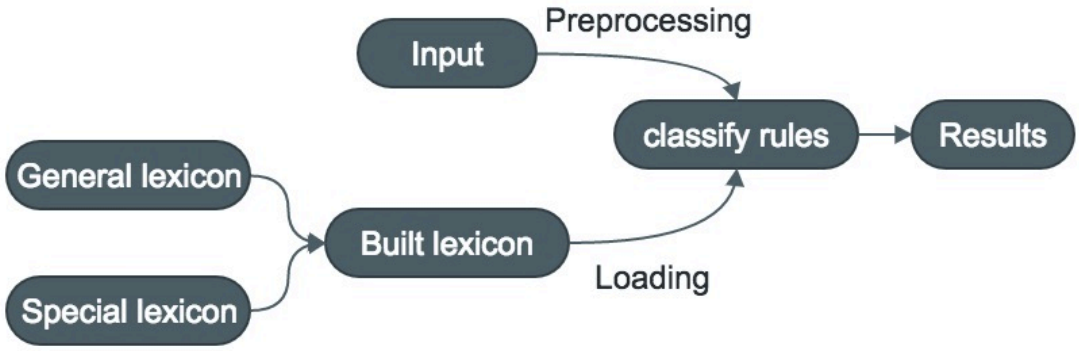


Fig. 2.1 Procedure of lexicon-based methods

2.1 Lexical resources

Since the classification results of this method is highly depended on the lexicon, we should be very careful when choose one. For English lexicon, most researchers built their dictionary based on the WordNet [3] from Princeton. But the dataset is too big and complicate for our case, so we didn’t use it as our fundamental lexical. Besides, Peter Turney applied PMI (Point of Mutual Information) to calculate the semantic orientation of words, phrases and text, which is one of the main approaches to the problem of sentiment analysis. [4]

In the beginning, we applied a general lexical SCL-NMA (Sentiment Composition Lexicon for Negators, Modals, and Degree Adverbs) [5]. But it turns out that the general domain lexical did not perform good. Therefore, in order to get a better performance, we try to build our own lexicon by combining the general lexical with the special field lexical respect to our test dataset, which are Amazon and Yelp Sentiment Lexicons [6] created by Kiritchenko, et al.

Firstly, we edit those lexicons and let them have the same format, which is

<term><tab><score>. Where <term> can be a unigram or a bigram. In this project we choose unigram term, because in our dataset there are a lot of phrases that don't match with the bigram term of the lexicon. If we use bigram, lots of term with a clear sentiment will not be recognized which would lead to the bad performance of the system. Another parameter is <score>, which is a real-valued sentiment score [7]:

score = PMI(w, POS) - PMI(w, NEG)

where PMI stands for Point-wise Mutual Information between a term w and the positive/negative class.

Secondly, since the mixed lexicon has some overlaps, we have to delete the repeated words and keep only one term of each word. Besides, we update the lexicon with some new and special words closely related with our dataset. Then adjust the weight of these new words to fit in the lexicon.

2.2 Classification

After we have the lexicon, we have a set of basic word with polarity and strength corresponding to each word. To extract the sentiment of a new comment, we compared each word in the comment with the lexicon, and get the score of each word. Then sum them up to get the whole value of the comment. If the sum of scores is over zero, we classify the comment as positive. If under zero, negative. But when it equals to zero, the system doesn't know how to classify it. Therefore, we randomly classify it to positive or negative categories. The equation can be denoted as follows:

$$\text{Polarity of comments} = \begin{cases} Positive \ \ (sum(scores) > 0) \\ Randomly\ classify \ \ (sum(scores) = 0) \\ Negative \ \ (sum(scores) < 0) \end{cases}$$

3. Supervised Classification

3.1 Feature

Our data is users' reviews from Internet, which are sentences. To classify the data with machine learning algorithms, the text data should be transformed into mathematical representation.

3.1.1 Feature selection

BOW (bag of words) approach is applied in this part, and specifically, we use Unigram, which is single word. Even though we remove the stop words with stop words list, the dimension of features is still large. To select the words which contain more sentiment information, we use Chi-square method, and select top 1500 words as feature items.

3.1.2 Weight

We use TF-IDF approach[8] to compute the weights of features and build the feature vector. TF is the frequency of each term, and IDF is inverse document frequency. With Chi-square and TF-IDF approaches, we transform each sentence to a feature vector. The dimension of the feature vector is 1×1500, and we notice that the feature vector is sparse.

3.2 Classification Models

We tried Naïve Bayes, KNN and Support Vector Machine for the classification.

3.2.1 Sentimental dictionary

find a word in emotion dictionary, and find the degree words (if any)before the emotion word and make a score. If there are any exclamation point at the end of sentence, double the score.

3.2.2 Naïve Bayes

Bayes classification is a fundamental and important tool in machine learning, and it is still widely used for solving text categorization problem nowadays. With the using of words frequency as features plus some appropriate pre-processing, Naïve Bayes is very competitive in this domain.

3.2.3 KNN

We realize KNN methods with python2.7. After several experiments, we choose K to be 20. Generally, KNN does not have a good performance on this classification task.

3.2.4 SVM

We use SVM lib from SKLEARN[9] under python 2.7. Because data are sparse, we choose linear Kernel. For the penalty parameter, we use default value which is 1. SVM performs well on this task.

3.2.5 Classification model for English reviews.

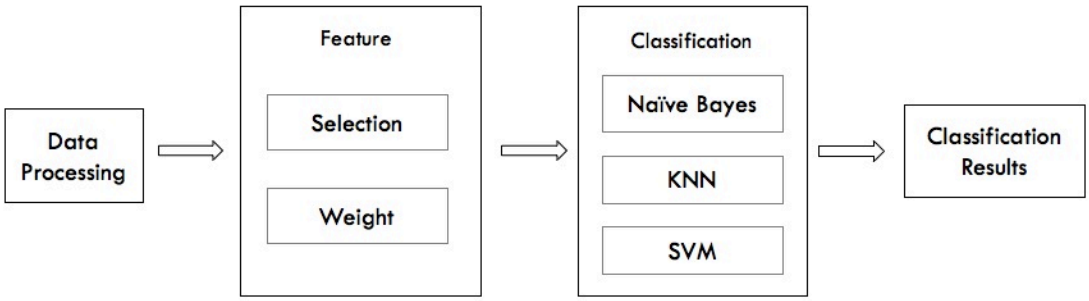


Fig. 3.1 Procedure of supervised methods for English reviews

4. Results

4.1 Sentiment classification based on lexicons

In this part, we try different lexicons on the system, and compute the accuracy:

Table 4.1 performance of different lexicon

Diff lexicon	SCL-NMA (general lexicon)	Amazon Sentiment Lexicon	Yelp Sentiment Lexicon	Our Combined Lexicon
Accuracy	65%	72%	77%	78%

4.2 Supervised Classification for English data

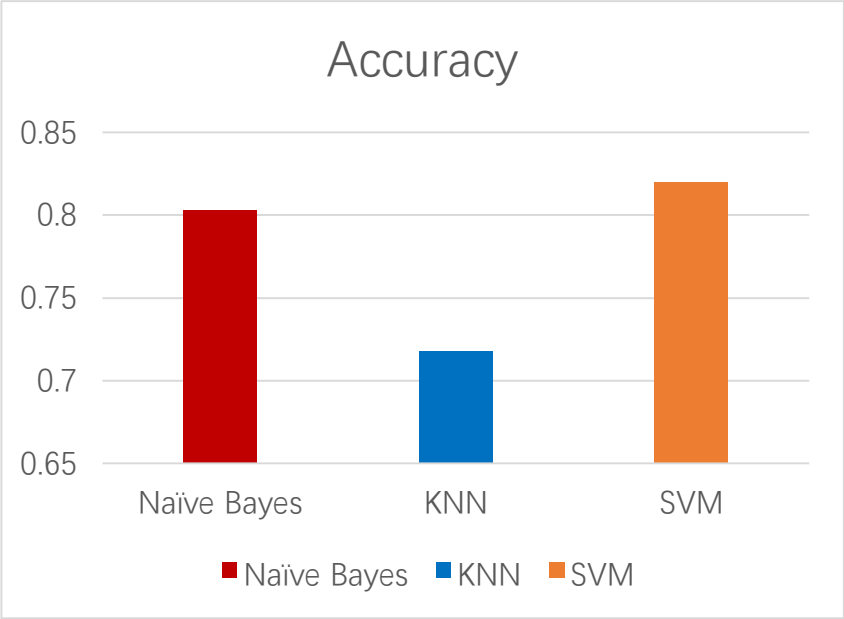
We use precision, recall, F1-score and accuracy to show our classification results. The specific results are listed below.

Table 4.2 classification result 1

Model	Precision	Recall	F-1 score	Accuracy
Naïve Bayes	POS: 0.84 NEG: 0.77	POS: 0.76 NEG: 0.85	POS: 0.80 NEG: 0.81	0.80
K-NN	POS: 0.77 NEG: 0.68	POS: 0.65 NEG: 0.79	POS: 0.70 NEG: 0.73	0.72
SVM	POS: 0.81 NEG: 0.84	POS: 0.86 NEG: 0.78	POS: 0.83 NEG: 0.81	0.82

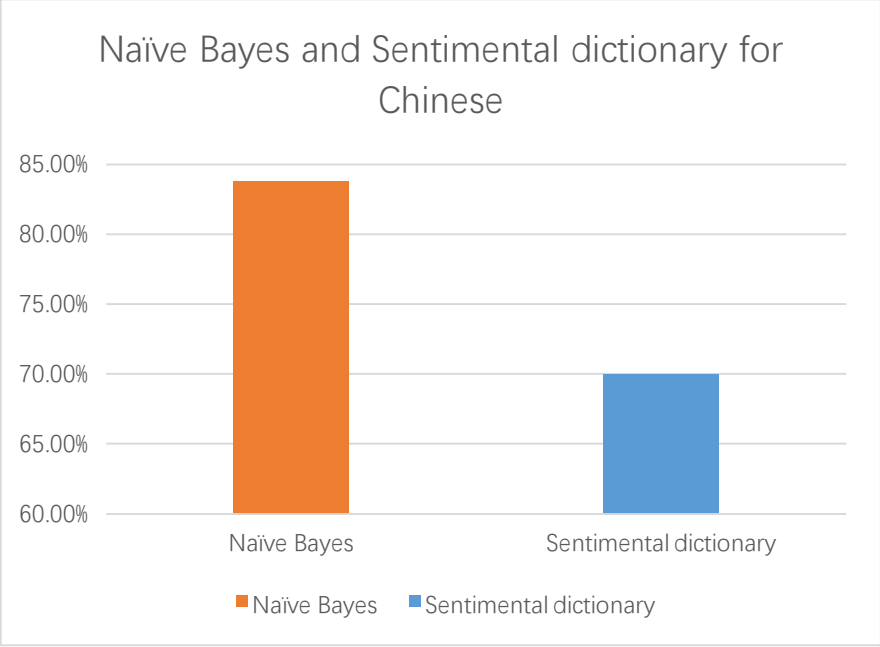
Also, we use histogram to show the accuracy:

Table 4.3 classification result 2



4.3 Sentiment classification for Chinese reviews using Naïve Bayes and sentimental dictionary

Table 4.4 classification result for Chinses text



5. Analysis

5.1 Lexicon-based methods for English text

Lexicon-based method is easy to implement and have quit good performance. However, it over relies on the quality and correlation of lexicons.
Also, building a lexicon which perfectly matches with our data would cost lots of human and material resources. Therefore, classification approaches using machine learning methods have more advantages.

5.2 Supervised Classification for English text

Our approaches combining Chi-square and TF-IDF with Naïve Bayes, KNN and SVM can finish the classification task. Using Chi-square and TF-IDF, we can transform the text data to sparse vectors. For classification, SVM achieve the best results with accuracy 0.82; the result of Naïve Bayes is not as good as SVM, but it is still satisfying with accuracy 0.80. However, KNN performs not very well with accuracy only 0.72.

Since we use BOW approach, which means we only focus on words, our approach cannot analyze the sentence structures. For some comments with turning, irony, double-negation, it is hard to classify this kind of reviews with high accuracy. Especially, when people express their feelings about things, they prefer to use negation expressions. So in the future work, the approach combining the information of the sentence structures should have better classification results.

5.3 Naïve Bayes and sentimental dictionary Result for Chinses text

In this experiments Naïve Bayes works better then sentimental dictionary. For Naïve Bayes Total accuracy is 86%. However there are some problems using naïve Bayes for text categorization. By using Naïve Bayes, we consider there are no relationship between each element(word) in a sentence and we omit the sentence structure in a sentence. This is an main drawback about using Naïve Bayes to do documents' classification. Because according to normal writing habit, word in a sentence have relationship with other words in the context more or less. The relatively high accuracy may be owed to that the training comments and testing comments are only associated with one product. If the comments are about other comments. The classifier may not work well. For sentimental dictionary, the total accuracy is 70%. The pros of sentimental dictionary is considering the relation between words and the structure of sentence. The cons of sentimental dictionary is relying too much on the occurrence of emotional words. For specific comments on given product without emotional words. It won't work.

6. Reference

[1]Taboada, Maite, et al. "Lexicon-based methods for sentiment analysis." Computational linguistics 37.2 (2011): 267-307.
[2] *https://archive.ics.uci.edu/ml/datasets/Sentiment+Labelled+Sentences#*
[3]Princeton University "About WordNet." WordNet. Princeton University. 2010. http://wordnet.princeton.edu
[4]Turney, Peter. 2002. Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. In Proceedings of 40th Meeting of the Association for Computational Linguistics, pages 417–424, Philadelphia, PA.
[5]Svetlana Kiritchenko and Saif M. Mohammad (2016) The Effect of Negators, Modals, and Degree Adverbs on Sentiment Composition. Proceedings of the 7th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis (WASSA), San Diego, California, 2016.
[6]Kiritchenko, S., Zhu, X., Cherry, C., and Mohammad, S. (2014) NRC-Canada-2014: Detecting Aspects and Sentiment in Customer Reviews. Proceedings of the 8th International Workshop on Semantic Evaluation Exercises (SemEval-2014), Dublin, Ireland, 2014.
[7] Kiritchenko, S., Zhu, X., Mohammad, S. (2014). Sentiment Analysis of Short Informal Texts. Journal of Artificial Intelligence Research, 50:723-762, 2014.
[8] http://scikit-learn.org/stable/index.html
[9] Aizawa A. An information-theoretic perspective of tf–idf measures[J]. Information Processing & Management, 2003, 39(1): 45-65.
[10] Liu B. Sentiment analysis and opinion mining[J]. Synthesis lectures on human language technologies, 2012, 5(1): 1-167.
[11] http://www.mindfuleye.com/about/lexant.htm
[12] Terveen et al.1997] Loren Terveen, Will Hill, Brian Amento, David McDonald, and Josh Creter. 1997. PHOAKS: A system for sharing recommendations. Communications of the ACM, 40(3):59–62.
[13] [Tatemura2000] Junichi Tatemura. 2000. Virtual reviewers for collaborative exploration of movie reviews. In Proc. of the 5th International Conference on Intelligent User Interfaces, pages 272–275

Division of work:

Zhe Hu: Supervised classification model for English text

Designed and realized English reviews classification with supervised approaches(Naïve Bayes, KNN and SVM), including:

1. Feature selection using BOW and Chi-square;
2. Using TF-IDF to compute feature weights, and transform sentence data to feature vectors;
3. Classification part: SVM, KNN and Naïve Bayes methods.
4. Analyze the results of supervised classification, and compare the performances of different classification methods.

Yu Yin: Unsupervised classification model for English text

1. Data cleaning, data preprocessing in English part.
2. Trying to select features from data through Information Gain(IG) method, but failed because we care more about the specific class not the whole system.
3. Sentiment analysis realization with lexicon-based methods.
4. Analyzing the results of lexicon-based methods, and discussing its cons and pros comparing to supervised methods.

Shaoyang Yan: Supervised classification model for Chinese text

1.Data cleaning, data preprocessing in Chinese part.
2.Using python to realize Naive Bayes on Chinese document classification
3.Using sentimental dictionary method to do document classification
4.Analyze and compare the results of the two methods